REMARKS

Applicant respectfully requests the Examiner's reconsideration of the present application as amended.

Claims 1-33 are pending in the present application.

Claims 1-33 are rejected under 35 U.S.C. §102(e) as being unpatentable over U.S. Patent No. 6,817,005 ("Mason").

Claims 1, 12, 19, and 25 have been amended.

Support for amended claims 1, 12, 19, and 25 is found in pages 4-29 of the specification, Figures 1-17 of the drawings, and claims 1-33 as originally filed. No new matter has been added.

Claims 1-33 are rejected under 35 U.S.C. §102(e) as being unpatentable over Mason. In particular, with respect to claims 1 and 25, the Examiner states that

> As to claims 1 and 25, Mason et al. teach a method for designing a system on a target device utilizing logic devices (PLD) (see Figs. 1-8 and its description, summary) comprising generating options for utilizing resources on the PLDs in response to user specified constraints (creating unelaborated modules, determining size, location of each unelaborated module to be implemented on PLDs corresponding to determining possible locations to place a component and move the component, user constraints); and refining the options (performing optimizations of placement and routing) for utilizing the resources on the PLDs independent of the user specified constraints (with constrainings and without user constraints, see col. 9; the floorplanner tool is used to create position and range (size and shape) constraints for each module, and locate top-level logic as well as module ports and any associated constrained, see at least see summary).

(1/11/2004 Office Action, pp. 2-3).

Applicants respectfully submit that Mason does not render claim 1 unpatentable under 35 U.S.C. §102(e).

Mason discloses a modular design flow where logic designers are able to partition a top-level logic design for a PLD into modules and implement any module independently from other modules. Modules are mapped, placed, and routed using selected information derived at the time the top-level logic design is partitioned. Finally,

the modules are integrated into the top-level logic design using a guided process. Specifically, the information generated during the partitioning of the top-level design and the implementation of each module is used to guide the implementation of the associated logic in the top-level design. In this manner, the implementation of all modules can proceed in any order of in parallel and the integration of the modules into the top-level design can be done quickly and in any order (Mason Abstract).

It is submitted Mason does not teach or suggest a method for designing a system on a target device utilizing PLDs that include generating options of utilizing resources on the PLDs in response to user specified constraints, and refining the options where the options are independent of the user specified constraints.

On the contrary, Mason discloses a constraint editor tool that is run on the top-level native generic database (NGD) file to generate a top-level user constraint file (UCF) (col. 2, line 66 to col. 3, line 1). In the module build stage, the top-level user constraint file is copied into a module director, creating a module-relative top-level UCF file. A module netlist is created using a standard HDL editor. The module is defined by a module level EDIF file and a module level netlist constraints file (NCF). The ngdbuild tool is run on the module level EDIF and NCF files as well as the module-relative top-level UCF file and the top-level NGO file to generate a module-level NGO file and a module-relative top-level NGD file (col. 3, line 61 through col. 4, line 3). In the module map stage, a mapper tool is run on the module-relative, top-level NGD file (col. 4, lines10-11). Mason fails to disclose generating options of utilizing resources on the PLDs in response to user specified constraints, and refining the options where the options are independent of the user specified constraints.

The Examiner states that Mason teaches "performing optimizations of placement and routing ... ... with constraints and without user constraints" and refers to column 9 for support (1/11/2004 Office Action, pp. 2-3). Applicants are unclear as to how column

9 supports the Examiner's position. Mason states in column 9, lines 55-58 that "The top.ngo file includes the top-level design with any constraints from the top.edn and top.ncf files, but without any constraints from a user constraints file." However, Mason then proceeds to describe that the top.ncf file is used together with the top.ngo file.

> In step 108, a command ngdbuild-modular initial top.ngo top.ucf triggers the ngdbuild tool to run on the top-level NGO (top.ngo) and the top-level UCF (top.ucf) file to write the constraints of the top-level UCF (top.ucf) file into the top-level NGD (top.ngd) file.

(Mason, col. 10, lines 14-19).

> This top-level UCF (top.ucf) file will be used in the module build stage 200 to correctly position each model. For this reason, the ngdbuild tool does not need to be run to reannotate these constraints back into the top-level NGD (top.ngd) file. Instead, the top.ngo and top.ucf files will be used directly in subsequent stages.

(Mason, col. 12, lines 27-33).

> In step 203, the ngdbuild tool is run on the top-level NGO (top.ngo) file (still located in the TOP/directory), the module level EDIF (A.edn) and NCF (A.ncf) files, and the module-relative top-level UCF file.

(Mason, col. 13, lines 1-4)

> Specifically, in step 801, command ngdbuimodular assembpim_path pim_path –use_pim mouse_pim mouse_pim mode top.ngo triggers the ngdbuild tool to run on the PIM level NGO (A.ngo, B.ngo, and C.ngo) files for each of the implemented modules as well as the top-level NGO (top.ngo) and UCF (top,ucf) files.

(Mason, col. 17, lines 6 through 12).

Furthermore, the "optimizations" disclosed in Mason do not describe generating options of utilizing resources on the PLDs in response to user specified constraints, and refining the options generated where the options are independent of the user specified constraints (see Mason column 3, lines 34-39, col. 12, lines 4-10, column 17, line 66 to column 18, line 11). Applicants respectfully request that the Examiner specifically point to the section in Mason which discloses performing optimizations of the utilization of resources during placement and routing with user specified constraints and performing

optimizations of the utilization of resources during placement and routing without the user specified constraints.

In contrast, claim 1 as amended states

> 1. A method for designing a system on a target device utilizing programmable logic devices (PLDs), comprising:
> generating options for utilizing resources on the PLDs in response to user specified constraints; and
> refining the options for utilizing the resources on the PLDs where the options are independent of the user specified constraints.

(Claim 1) (Emphasis added).

Claims 12, 19, and 25, as amended, include similar limitations. Given that claims 2-11, 13-18, 20-24, and 26-33, depend directly or indirectly from claims 1, 12, 19, and 25, as amended, it is likewise submitted that claims 2-11, 13-18, 20-24, and 26-23 are also patentable under 35 U.S.C. §102(e) over Mason.

In addition, Applicants submit that Mason also fails to disclose refining the options for utilizing the resources where the refining is performed in response to the options not satisfying design parameters or the options not satisfying the user specified constraints.

The Examiner states

> As to claims 2-3, 16-17 and 26-27, Mason et al. teach refining the options for utilizing the resources is performed in response to the options not satisfying design parameters (performing optimizing of modules placement and routing, modifications allowing fine-tuning the design to meet timing constraints; at least see col. 12-14).

(1/11/2005 Office Action, p. 3).

Applicants submit that columns 12 through 14 describe a module build stage and a module map stage. Columns 12 through 14 do not, however, describe generating options of utilizing resources on the PLDs in response to user specified constraints, and refining the options where the options are independent of the user specified constraints, wherein the refining is performed in response to the options not satisfying design

parameters or the options not satisfying user specified constraints. In fact, column 13 of Mason discloses that user specified constraints are used in the model build stage. "The copied UCF file top.ucf in the current directly is explicitly specified in the argument to the command." (col. 13, lines 22-24).

In contrast, claims 2 states

> 2. The method of Claim 1, wherein refining the options for utilizing the resources is performed in response to the options not satisfying design parameters.

(Claim 2).

Claims 16, 23, and 26 include similar limitations.

Claim 3 states

> 3. The method of Claim 1, wherein refining the options for utilizing the resources is performed in response to the options not satisfying the user specified constraints.

(Claim 3).

Claims 17, 22, and 27 include similar limitations.

Furthermore, Applicants submit that Mason also fails to disclose refining the options for utilizing the resources where the refining is performed in response to a threshold number of options generated.

The Examiner states that

> As to claims 4, 18, 24, and 28, Mason et al. teach multiple guides used in placer and router tool, where the placer and router tool iteratively steps through each one and guides the placement and routing of logic corresponding to the logic in the guide file (col. 22). Since the guide file provides complete information to perform placing and routing including optimization, the guide file must include a threshold number of options generated/number of possible locations/number of routing strategies in order to quickly implementing a final product according to user defined constraints.

(1/11/2005 Office Action, pp. 3-4).

Applicants submit that the "multiple guides used" referred to by the Examiner utilize user specified constraints. Mason discloses that "any constraints which are found

in the top-level NGD or UCF files are 'pushed' into these guided pieces of logic in a standard manner." (col. 22, lines 31-33). Applicants further submit that Mason fails to disclose that the "multiple guides used" operate in response to a threshold of options generated.

In contrast, claims 4 states

> 4. The method of Claim 1, wherein refining the options for utilizing the resources is performed in response to having a threshold number of options generated.

(Claim 4)

Claims 18, 24, and 28 include similar limitations.

Moreover, Applicants submit that Mason also fails to disclose refining the options for utilizing the resources where the refining is performed in response to a triggering event.

The Examiner states that

> As to claims 5 and 29, Mason et al. also teaches triggering event (col. 9, col. 13, col. 17).

(1/11/2005 Office Action, p. 4).

Applicants submit that although columns 9, 13, and 17 include the term "triggers", the term does not refer to an event that refines options for utilizing resources where the options are independent of user specified constraints. For example, column 13 of Mason discloses that user specified constraints are used in the model build stage. "The copied UCF file top.ucf in the current directly is explicitly specified in the argument to the command." (col. 13, lines 22-24). Also, column 17 of Mason discloses commands that triggers the ngdbuild tool to run on the PIM level NGO files for each of the implemented "as well as the top-level NGO (top.ngo) and UCF (top.ucf) files." (col. 17, lines 7-12).

In contrast, claims 5 states

> 5. The method of Claim 1, wherein refining the options for utilizing the resources is performed in response to a triggering event.

(Claim 5)

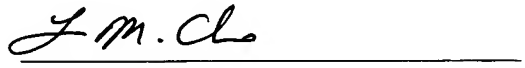
Claims 29 includes similar limitations.

In view of the amendments and arguments set forth herein, it is respectfully submitted that the applicable rejections and have been overcome. Accordingly, it is respectfully submitted that claims 1-33, as amended, should be found to be in condition for allowance.

If any additional fee is required, please charge Deposit Account No. 50-1624.

Respectfully submitted,

Dated: April 8, 2005



Lawrence M. Cho
Attorney for Applicant
Registration No. 39,942

P.O. Box 2144
Champaign, IL 61825
(217) 377-2500